

1 1. A method comprising:
2 initializing to false a predicate that guards a speculative instruction in a software-
3 pipelined loop;
4 executing at least one iteration of the software-pipelined loop, including an
5 instruction that sets the predicate to true if an associated live-in value is consumed; and
6 executing the speculative instruction in subsequent iterations of the software-
7 pipelined loop.

1 2. The method of claim 1, wherein the instruction that sets the predicate true is gated by a
2 stage predicate of the software-pipelined loop.

1 3. The method of claim 2, wherein executing - at least one iteration of the software-
2 pipelined loop comprises executing the predicate setting instruction when the stage predicate is
3 true.

1 4. The method of claim 2, wherein the stage predicate is selected to delay execution of the
2 speculative instruction until the live-in value has been consumed.

1 5. The method of claim 1, wherein initializing to false a predicate comprises initializing to
2 false a predicate other than a stage predicate.

1 6. A method comprising:
2 initializing a software-pipelined loop to deactivate a speculative instruction;
3 executing at least one initiation interval (II) of the software-pipelined loop;
4 activating the speculative instruction; and
5 executing subsequent IIs of the software-pipelined loop.

1 7. The method of claim 6, wherein initializing the software-pipelined loop comprises
2 initializing as false a predicate that guards the speculative instruction.

1 8. The method of claim 7, wherein executing at least one II of the software-pipelined loop
2 comprises executing an instruction that determines a value for the predicate guarding the
3 speculative instruction.

1 9. The method of claim 8, wherein activating the speculative instruction comprises
2 executing the speculative instruction if the predicate is true.

1 10. The method of claim 6, wherein the speculative instruction is a compare instruction and
2 initializing the software pipeline to deactivate the speculative instruction comprises initializing a

3 rotating source register for the compare to a value for which a predicate determined by the
4 compare instruction is false.

1 11. The method of claim 10, wherein activating the speculative instruction comprises rotating
2 a value into the source register used by the compare to determine if the predicate is true.

1 12. The method of claim 7, wherein executing at least one II of the software-pipelined loop
2 comprises executing an instruction that activates the speculative instruction.

1 13. A method for software pipelining a “while” loop comprising:
2 identifying a speculative instruction in the loop;
3 guarding the speculative instruction with a sticky predicate;
4 initializing the sticky predicate to false; and
5 inserting an instruction to set the sticky predicate true at a specified initiation
6 interval of the loop.

1 14. The method of claim 13, wherein inserting an instruction comprises an instruction to set
2 the sticky predicate true when a live-in value targeted by the speculative instruction is consumed.

1 15. The method of claim 10, wherein the inserted instruction is a compare instruction that is
2 gated by a stage predicate.

1 16. The method of claim 15, wherein the inserted instruction evaluates the sticky predicate as
2 true when it is gated on by the stage predicate.

1 17. The method of claim 16, wherein the stage predicate is selected to activate the inserted
2 instruction once the live-in value is consumed.

1 18. An apparatus comprising a machine readable medium on which are stored instructions
2 that may be executed by a processor to implement a method comprising:

3 executing a stage of a software-pipelined loop that includes a speculative
4 instruction, the speculative instruction being gated off by a sticky predicate;

5 executing an instruction that sets the sticky predicate; and

6 executing the stage of the software-pipelined loop, including executing the
7 speculative instruction.

1 19. The machine-readable medium of claim 18, wherein the method further comprises
2 initializing the sticky predicate to false to gate the speculative instruction off prior to executing
3 the software-pipelined loop.

1 20. The machine-readable medium of claim 18, wherein executing an instruction that sets the
2 sticky predicate comprises:

3 rotating a new value into a stage predicate that guards the sticky predicate setting
4 instruction; and

5 executing the sticky predicate setting instruction when the stage predicate is true.

1 21. A computer system comprising:

2 a processor to execute instructions; and

3 a memory to store instructions which may be executed by the processor to
4 implement a method comprising:

5 executing an initiation interval of a software-pipelined loop that includes a
6 speculative instruction, the speculative instruction being gated off by a sticky
7 predicate;

8 executing an instruction that sets the sticky predicate; and

9 executing a subsequent initiation interval of the software-pipelined loop,
10 including executing the speculative instruction.

1 22. The computer system of claim 21, wherein the method further comprises initializing the
2 sticky predicate to false to gate the speculative instruction off prior to executing the software-
3 pipelined loop.

1 23. The computer system of claim 22, wherein executing an instruction that sets the sticky
2 predicate comprises:
3 rotating a new value into a stage predicate that guards the sticky predicate setting
4 instruction; and
5 executing the sticky predicate setting instruction when the stage predicate is true.

1 24. A computer system comprising:
2 a processor to execute instructions; and
3 a memory to store instructions which may be executed by the processor to:
4 initialize a software-pipelined loop to deactivate a speculative instruction;
5 execute at least one initiation interval (II) of the software-pipelined loop;
6 activate the speculative instruction; and
7 execute subsequent IIs of the software-pipelined loop.

1 25. The computer system of claim 24, wherein the processor initializes the software-
2 pipelined loop by at least initializing as false a predicate that guards the speculative instruction.

1 26. The computer system of claim 25, wherein the processor executes at least one II of the
2 software-pipelined loop by at least executing an instruction that determines a value for the
3 predicate guarding the speculative instruction.